Binomial tau-leap spatial stochastic simulation algorithm for applications in chemical kinetics

Tatiana T. Marquez-Lago and Kevin Burrage

Citation: The Journal of Chemical Physics **127**, 104101 (2007); doi: 10.1063/1.2771548

View online: http://dx.doi.org/10.1063/1.2771548

View Table of Contents: http://scitation.aip.org/content/aip/journal/jcp/127/10?ver=pdfcov

Published by the AIP Publishing

Binomial tau-leap spatial stochastic simulation algorithm for applications in chemical kinetics

Tatiana T. Marquez-Lago[a] and Kevin Burrage[b]
Advanced Computational Modeling Centre, The University of Queensland, Brisbane QLD 4072, Australia

(Received 12 June 2007; accepted 23 July 2007; published online 10 September 2007)

In cell biology, cell signaling pathway problems are often tackled with deterministic temporal models, well mixed stochastic simulators, and/or hybrid methods. But, in fact, three dimensional stochastic spatial modeling of reactions happening inside the cell is needed in order to fully understand these cell signaling pathways. This is because noise effects, low molecular concentrations, and spatial heterogeneity can all affect the cellular dynamics. However, there are ways in which important effects can be accounted without going to the extent of using highly resolved spatial simulators (such as single-particle software), hence reducing the overall computation time significantly. We present a new coarse grained modified version of the next subvolume method that allows the user to consider both diffusion and reaction events in relatively long simulation time spans as compared with the original method and other commonly used fully stochastic computational methods. Benchmarking of the simulation algorithm was performed through comparison with the next subvolume method and well mixed models (MATLAB), as well as stochastic particle reaction and transport simulations (CHEMCELL, Sandia National Laboratories). Additionally, we construct a model based on a set of chemical reactions in the epidermal growth factor receptor pathway. For this particular application and a bistable chemical system example, we analyze and outline the advantages of our presented binomial τ-leap spatial stochastic simulation algorithm, in terms of efficiency and accuracy, in scenarios of both molecular homogeneity and heterogeneity. © *2007 American Institute of Physics*. [DOI: 10.1063/1.2771548]

INTRODUCTION

A deterministic temporal approach for simulating cell signaling chemical pathways is only adequate when the reactions involved have both large numbers of reactant molecules and when discreteness and internal noise have no noticeable macroscopic effects. Hence, if either the number of molecules of some species is small *or* the system is susceptible to noise amplification, as often happens in a cell, one has to account for the discrete and stochastic nature of the system. This can be captured by evolving a discrete nonlinear Markov process that has a probability density function that is the solution of the so-called chemical master equation (CME), a discrete parabolic partial differential equation in which there is an equation for each configuration of the state space. The CME can rarely be solved analytically and can be very computationally demanding due to the possibly large numbers of different reactant molecular species considered and the generally high nonlinear nature of the pathway, conditions that are regularly encountered within cell signaling systems.[1] In consequence, researchers started to work out ways around the inherent difficulties involved in solving the CME and so, a little more than 30 years ago, kinetic Monte Carlo (KMC) algorithms started to be developed. In 1966, Young and Elcock published the first basic features of the KMC method, followed in 1975 by Bortz, Kalos, and Lebowitz, with the development of the *n-fold way* (also known as BKL), a KMC algorithm simulating the Ising model.[2] In the following year, Gillespie coined the term SSA, stochastic simulation algorithm, to describe a KMC method describing chemical kinetic evolution in time.[3] For simplicity, we will use the common terminology of SSA, but it should be noted that the algorithm and the time advancement scheme of the SSA (Ref. 3) are essentially the same as in BKL.[2]

The SSA is an exact procedure that describes the evolution of a discrete nonlinear Markov process. It accounts for the internal stochasticity of m reactions and only updates the number of molecules inside the system in integer numbers, a characteristic that better resembles the molecular biology. In brief, the SSA simulates two random numbers from the uniform distribution in the unit interval, at each step, to evaluate an exponential waiting time τ, for the next reaction to occur and an integer j between 1 and m that indicates which reaction occurs based on the relative sizes of the propensity functions, i.e., the probabilities of the reactions. The state vector $X(t)$ contains the number of molecules of each different species and is updated at the new time point by the addition of the jth stoichiometric vector to the previous value of the state vector, that is, $X(t+\tau)=X(t)+\nu_j$.

We note that there is an alternative implementation to the SSA, the next-reaction method[4] developed by Gibson and Bruck, on which the NSM is based. However, in either implementation, the time step can become very small, especially if there are large numbers of molecules and/or widely varying rate constants. In order to overcome these limita-

[a]Author to whom correspondence should be addressed. Electronic mail: tmarquez@maths.uq.edu.au
[b]Electronic mail: kb@maths.uq.edu.au

tions, a number of different coarse-graining techniques have been created in views of reducing overall computational costs, where a much larger time step can be used at the loss of a small amount of accuracy. An example of these are the so called τ-leap methods, in which the sampling of likely reactions to happen inside a possibly longer step of size τ is drawn from either a Poisson[5,6] or binomial[7] distributions.

Very briefly, the update procedure for the Poisson τ-leap method can be written as $X(t+\tau)=X(t)+\Sigma_{j=1}^{M}\nu_j K_j$, where $K_j=P(a_j(X(t))\tau)$ for reactions $j=1,\ldots,M$ is a sample from the Poisson distribution with mean $a_j(X(t))\tau$.[5] Here $a_j(X(t))$ is the propensity function for the jth reaction. The binomial τ-leap method has a similar update formula but now the various K_j take the form $K_j=B(N_j,P_j)$, where there are some subtleties in the form of the N_j and P_j,[7] and the variables N_j and P_j represent the sample size and probability of occurrence of reaction type j, respectively. In either case, the τ step is controlled by a selection strategy that depends on a prespecified control parameter ε, such that $0<\varepsilon\ll 1$.

By choosing an appropriate value of ε one can ensure that the relative changes in all propensity functions over the τ step are small and, so, a greater accuracy in the approximation can be attained. However, there is a trade-off between accuracy and CPU time, i.e., smaller values of ε result in longer simulation times.

A further, yet natural, complication is that biological systems are in many cases characterized by complex spatial structure, low diffusion rates, and low numbers of molecules, hence requiring spatially resolved simulations. The most straightforward technique is the deterministic approach, i.e., via the reaction-diffusion partial differential equation, an approach that is only valid if dealing with large molecular concentrations and when noise is not amplified throughout the system. If at least one of these conditions fails to hold, one must rely on spatial stochastic simulators, which can be of discrete or continuous nature and with different levels of spatial resolution.

The highly resolved end of the discrete spatial stochastic simulators spectrum is represented by lattice and off-lattice particle based methods. Examples of the first are Monte Carlo simulators, in which a two dimensional or three dimensional computational lattice is used to represent a membrane or the interior of some part of a cell.[8–10] In these approaches, the lattice is "populated" with particles of different molecular species that may diffuse throughout the simulation domain and, depending on user-specified reactions, appropriate chemical reactions can take place with a certain probability. A different approach to simulating chemical pathways with particles are off-lattice simulators, in which the domain is discretized to efficiently localize particles, but each particle has a single reaction bin size that depends on its particular diffusion rate. If one or more molecules happen to be inside such a bin, appropriate chemical reactions can take place with a certain probability, and if a reaction is readily performed, the reactant particles are flagged. Some examples of these types of simulators are CHEMCELL (Refs. 11 and 12) and MCELL.[13]

Needless to say, particle methods can provide very detailed simulations of highly complex systems at the cost of exceedingly large amounts of computational time and, possibly, restrictions on the size of the simulation domain. In consequence, such detailed simulations can often only yield short simulation time spans that may not be of any interest to the experimentalists. Hence, it is important to remember that there is a trade-off between simulation time and resolution, as is illustrated in Fig. 1.

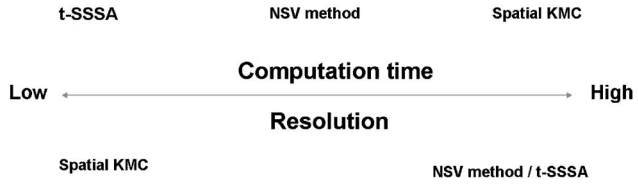

FIG. 1. Trade-off between simulation time and resolution.

The aim of this paper is to provide a spatially resolved method that yields accurate spatial chemical kinetics in meaningful simulation times that are of actual biological interest. We have achieved this by coarse graining a modified version of an existing spatial stochastic simulation algorithm commonly known as the next subvolume method,[14–16] which out of simplicity we will refer to as NSM. The NSM is a generalization of the SSA, where the volume is divided into separate subvolumes (SVs), that are small enough to be considered homogeneous by diffusion over the time scale of the reaction. At each step, the state of the system is updated by performing an appropriate reaction or by allowing a molecule to jump at random to a neighboring SV, with diffusion being modeled by a unary reaction. The expected time for the next event is only recalculated for those SVs that were involved in the current time step. The times for the next events are calculated similarly to the SSA algorithm (including a propensity function for diffusion events) and ordered in an event queue.

TAU SPATIAL STOCHASTIC SIMULATION ALGORITHM

Our presented algorithm, Bτ-SSSA, applies binomial tau leaping to a modified version of the original NSM method,[14–16] accounting for several reaction and/or diffusion

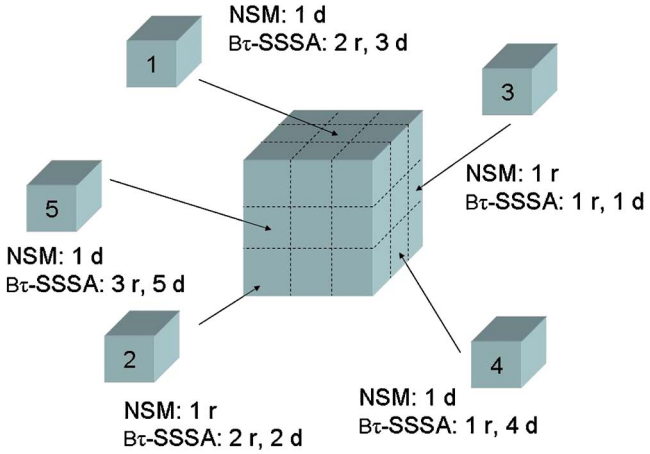

FIG. 2. Illustration of events within a time step in $B\tau$-SSSA as compared with NSM (r stands for reaction event and d for diffusion event).

TABLE I. Description of the matrices and vectors used in the NSM method and $B\tau$-*SSSA*.

N_V	Number of subvolumes
N_R	Number of possible reactions within the system
N_S	Number of different molecular species
ε	Error approximation parameter
D	$N_S \times 1$ vector containing the molecular diffusion rates of each species
K	$N_R \times 1$ vector containing the reaction rate constants
N matrix	$N_V \times 6$ matrix that specifies, for each SV, the six possible neighboring SVs to which molecules can jump by diffusion (up, down, left, right, in front, and behind)
C matrix	$N_V \times N_S$ matrix that specifies the number of molecules of each species contained in each SV
R matrix	$N_V \times 3$ matrix that contains, for each SV, the sums of reaction propensities $r_j=\sum_{i=1}^{N_V} a_{ij}$ (first column), the sum of diffusion propensities $s_j=\sum_{i=1}^{N_V} d_{ij}$ (second column), and r_j+s_j (third column)
Q matrix	$N_V \times 2$ matrix that contains the SV's ID (first column) and the time in which the next set of events inside that SV will take place (second column). The matrix is ordered with respect to increasing times
V matrix	$N_S \times N_R$ matrix, containing the stoichiometric vectors v
F matrix	$(N_R+N_S)\times(N_R+N_S)$ matrix, with the following nonzero entries: $f_{jk}=\sum_{i=1}^{N_R}\frac{\partial a_j(x)}{\partial x_i}\nu_{ik}$ for $j,k=1,\ldots,N_R$ and $g_{jk}=\sum_{i=1}^{N_S}\frac{\partial d_j(x)}{\partial x_i}e_{ik}$ for $j,k=1,\ldots,N_S$
M	$(N_R+N_S)\times 1$ vector, with $\mu_j=\sum_{i=1}^{N_R} f_{jk}(x)a_j(x)$ for $j=1,\ldots,N_R$ and $\mu_j=\sum_{i=1}^{N_R} g_{jk}(x)d_j(x)$ for $j=N_{R+1},\ldots,N_{R+S}$
Σ	$(N_R+N_S)\times 1$ vector, with $\sigma_j=\sum_{i=1}^{N_R} f_{jk}^2(x)a_j(x)$ for $j=1,\ldots,N_R$ and $\sigma_j=\sum_{i=1}^{N_R} g_{jk}^2(x)d_j(x)$ for $j=N_{R+1},\ldots,N_{R+S}$
T	$N_V \times 1$ vector, with $\tau_j=\min_k\left[\frac{\varepsilon(r_j+s_j)}{\lvert\mu_k\rvert},\frac{\varepsilon^2(r_j+s_j)^2}{\lvert\sigma_k\rvert}\right]$, where $k\in[1,N_R+N_S]$ and $j\in[1,N_V]$

TABLE II. Reaction propensities (diffusion propensities are calculated the same way as unary reaction propensities).

Reaction	Propensity	Stoichiometric coefficients
First order $S_k \xrightarrow{c_j} S_l$	$a_j=c_j X_k$	$\nu_{j,k}=-1$, $\nu_{j,l}=1$
Heterodimeric $S_k+S_l \xrightarrow{c_j} S_m$	$a_j=c_j X_k X_l$	$\nu_{j,k}=\nu_{j,l}=-1$, $\nu_{j,m}=1$
Homodimeric $S_k+S_k \xrightarrow{c_j} S_l$	$a_j=c_j X_k(X_k-1)/2$	$\nu_{j,k}=-2$, $\nu_{j,l}=1$
Hill type $S_k \xrightarrow{c_j} S_k+S_l$	$a_j=c_j f(X_k)$ where $f(X_k(t))=1-\frac{1}{1+(X_k(t)/X_0)^h}$ (activation) $f(X_k(t))=\frac{1}{1+(X_k(t)/X_0)^h}$ (inhibition), h describes the binding cooperativity and X_0 is such that $f(X_0)=1/2$.	$\nu_{j,l}=1$

events during each τ step, as is illustrated in Fig. 2. During each step, a SV is chosen according to increasing joint reaction-diffusion propensities, but if the calculated τ is less than a user-specified threshold, a modified version of a single step of the NSM is performed.

The description of the matrices and vectors used in the NSM (original and modified versions) and their tau-leap implementation can be found in Table I. It should be noted that each reaction propensity a_{ij} is calculated, as shown in Table II, and diffusion propensities are calculated the same way as unary reaction propensities. Additionally, the diffusion parameter d used to calculate s_j is equal to the experimental diffusion coefficient D divided by the length of a side of a SV.[15] In this way, as the domain is divided into more SVs the parameter d increases, reflecting the dominance of diffusion over reaction events.

Initialization.

(1) Set the following variables to zero: time, reaction counter, diffusion counter.

(2) Input reaction rates, diffusion rates, and stoichiometric vectors.

(3) Input the C matrix, i.e., the concentration of each molecular species in each SV.

(4) Generate the N matrix, defining the six possible neighbors for each SV, in accordance to selected boundary conditions.

(5) Create initial matrices R (containing reaction and diffusion propensities for each SV) and Q (SV time queue). It is advisable to set a maximal event time in the case of any SV containing no molecules [e.g., $Q(\mathrm{SV},2)$ =realmax, in MATLAB].

(6) Sort the SV's ID numbers according to increasing event time.

Iterations.

(7) Choose the SV with least event time, i.e., $Q(1,1)$.

(8) Draw a new uniformly distributed random number.

(9) If the total number of molecules inside this SV is larger than 2, proceed to step 10. Otherwise, automatically opt for a single event and, with r_{SV} representing the reaction propensity in SV, and s_{SV} the diffusion propensity in SV:

   (a) If $r_{\mathrm{SV}}=0$ perform one diffusion step, i.e., go to step 15.

   (b) If $r_{\mathrm{SV}}>0$ draw a new random number ξ, such that if $\xi<r_{\mathrm{SV}}/(r_{\mathrm{SV}}+s_{\mathrm{SV}})$ a reaction event will take place, in which case go to step 14. Otherwise, a diffusion event will take place, in which case go to step 15.

(10) Calculate an initial τ for this particular SV. This depends on the matrix F and vectors **M**, **Σ**, **T**, as defined in Table I. The initial τ is the minimum of the two entries in vector **T**.

(11) A minimal τ will be chosen, such that inside the selected SV, for all molecular species $j,k=1,\ldots,N_S$ and for all reactions in which this species is involved, τ complies with the restriction $P_j\tau\leqslant 1$, where $P_j=M_j/N$.

   (a) $M_j=a_j+d_k+d_m$, for a heterodimer reaction involv-

ing species k and m, in which case $N_j = \min[C(\mathrm{SV},k), C(\mathrm{SV},m)]$.

(b) $M_j = a_j + 2 \cdot d_k$, for a homodimer reaction involving species k, in which case $N_j = \lfloor C(\mathrm{SV},k)/2 \rfloor$.

(c) $M_j = a_j + d_k$, for a unary or Hill-type reaction involving species k, in which case $N_j = C(\mathrm{SV},k)$.

(d) A combination of any of the above. This is the case when any particular species involves more than one reaction, in which case the restriction is $\tilde{P}_j \tau \leqslant 1$, where $\tilde{P}_j = \tilde{M}_j / \tilde{N}_j$ and the $\tilde{M}_j$ will be a sum of as many corresponding M_j terms, as defined above, reflecting a reaction subnetwork within the system. On the other hand, $\tilde{N}_j$ will be the minimum of all corresponding N_j terms.

(12) Determine whether the step contains one event or multiple ones. If $r_{\mathrm{SV}} > 0$ then

(a) if τ is smaller than $\rho Q(1,2)$, where ρ is a user-specified natural number representing the lower limit on the number of events within a τ step, go to step 9(b).

(b) If τ is larger than $\rho Q(1,2)$, $\rho \geqslant 1$, go to step 15.

(13) Single chemical reaction event.

(a) Draw a new random number to sample which specific chemical reaction $j \in [1, N_R]$ occurred inside SV, according to $P = a_j(\mathrm{SV})/r_{\mathrm{SV}}$, the probability of occurrence of each reaction.

(b) Update the elements in SV (within the C matrix) by summing the corresponding stoichiometric vector corresponding to reaction j.

(c) Increase the time variable t by $Q(1,2)$ and the reaction counter by 1.

(d) Recalculate r_{SV} and s_{SV} and generate a new uniformly distributed random number ξ to update the Q matrix by $Q(\mathrm{SV},2) = 1/(r_{\mathrm{SV}} + s_{\mathrm{SV}}) \log(1/\xi)$, which will indicate the order related to the time of the next even in SV.

(e) Sort the SV numbers according to increasing event time.

(f) Go back to step 1.

(14) Single diffusion event.

(a) Draw a new random number and sample which species $s \in [1, N_s]$ will undergo a jump from SV to a contiguous randomly selected neighbor, according to the probability $P = d_s(\mathrm{SV})/s_{\mathrm{SV}}$.

(b) Draw a second uniformly distributed random number for selecting a new $\mathrm{SV}_{\mathrm{new}}$, which is one of the six neighbors of SV inside the N matrix.

(c) Update the C matrix by subtracting one molecule of species s from the elements in SV, and by adding one molecule of species s to the elements in $\mathrm{SV}_{\mathrm{new}}$, respectively.

(d) Increase the time variable t by $Q(1,2)$ and the diffusion counter by 1.

(e) Recalculate r_j and s_j for both $j = \mathrm{SV}$, $\mathrm{SV}_{\mathrm{new}}$ and generate two new uniformly distributed random numbers (ζ, ξ) to update the Q matrix as $Q(\mathrm{SV},2) = 1/(r_{\mathrm{SV}} + s_{\mathrm{SV}}) \log(1/\xi)$, $Q(\mathrm{SV}_{\mathrm{new}},2) = 1/(r_{\mathrm{SV}_{\mathrm{new}}} + s_{\mathrm{SV}_{\mathrm{new}}}) \log(1/\zeta)$.

(f) Sort the SV numbers according to increasing event time.

(g) Go back to step 1.

(15) τ-step.

For every single type of reaction, $j = 1, \ldots, N_R$, do the following:

(a) Determine $N_{j_{\mathrm{total}}}$, which represents the total numbers of reaction (type j) and diffusion events (of reactant species involved in reaction type j) to be performed during time τ. $N_{j_{\mathrm{total}}}$ will be sampled from a binomial distribution with parameters N_i and $P_j \tau$, where the latter is defined using the same procedure explained in step 11. Correspondingly, the number of reaction events of type j will be sampled from the binomial distribution with parameters $N_{j_{\mathrm{total}}}$ and a_j/M_j, and the number of diffusion events of each involved species s will be sampled from a binomial with parameters $N_{j_{\mathrm{res}}}$ and d_s/M_j, where the first parameter denotes the remaining molecules after the previous reaction and diffusion events calculated for this particular type of reaction.

(b) Sum the reaction and diffusion events and increase their counters correspondingly. Increase the time variable by τ.

(c) Update reaction events in the C matrix by summing the stoichiometric vectors multiplied by the number of reactions obtained in step 15(a).

(d) Update diffusion events in the C matrix. For each molecular species involved in reaction type j, subtract as many molecules as the number of diffusion events obtained for that particular species in step 15(a), adding them to the new contiguous SV's (selected from the N matrix).

(e) Recalculate the reaction and diffusion propensities $(r_{\mathrm{SV}}, s_{\mathrm{SV}})$.

(f) Update the Q matrix.

(g) Sort the SV numbers according to increasing event time.

(h) Go back to step 8.

ADDITIONAL IMPLEMENTATION ISSUES

Other than all the steps involved in the construction and implementation of tau leaping in time, we found that a binary tree is not necessarily effective for sorting the events. We have tested this approach, as well as sorting linearly the first $\hat{q}$ entries of the Q array, every $\tilde{q}$ number of steps, where $\hat{q} \gg \tilde{q}$. The latter approach can sometimes reduce computational time significantly but should be used carefully.

Of most importance, we found that, in order to match temporal dynamics using different numbers of compart-

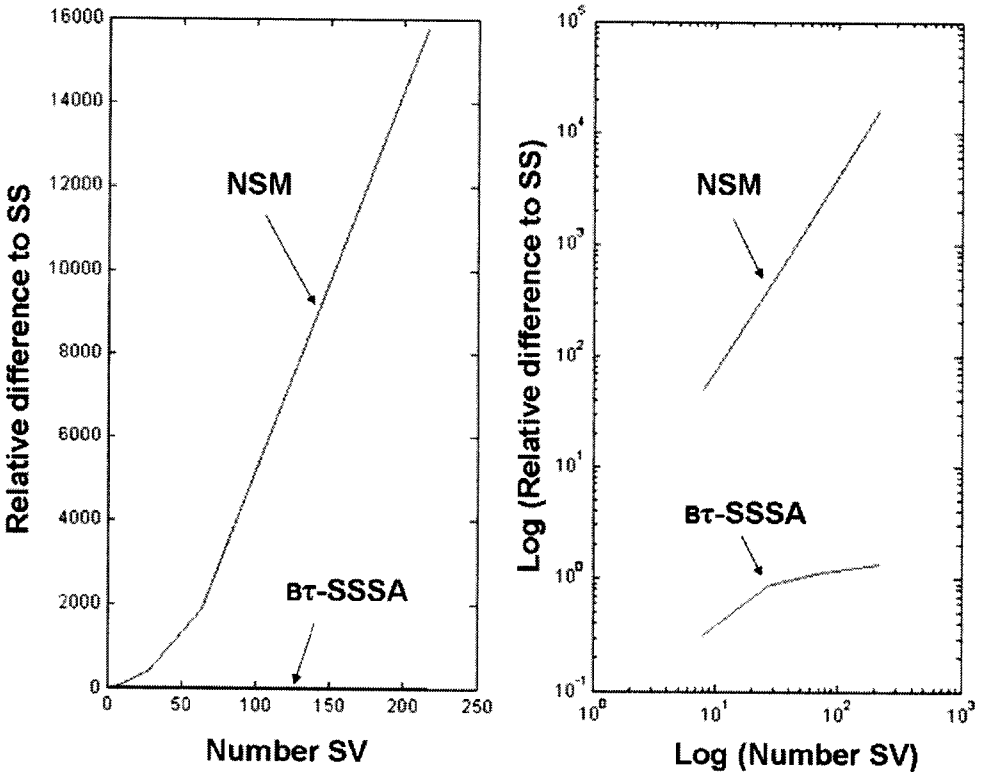

FIG. 3. Relative difference in time to reach a chemical steady state: Different numbers of subvolumes, with and without time scaling.

ments, we must rescale the diffusion and reaction propensities accordingly. It is easiest if we illustrate this with the following example.

If we consider a single fixed volume in which reactions $A+B\xrightarrow{k_1}C$ and $C\xrightarrow{k_2}D$ can happen, the sum of the reaction propensities and the sum of the diffusion propensities would be $a_0=k_1[A][B]+k_2[C]$ and $d_0=d_A[A]+d_B[B]+d_C[C]+d_D[D]$.

However, if we divide this volume into λ compartments, keeping the same concentration as a whole as in the original volume, we obtain for each SV a modified version of the reaction propensities and diffusion propensities, i.e., $\widetilde{a_{\mathrm{SV}}}=k_1[\widetilde{A}][\widetilde{B}]+k_2[\widetilde{C}]$ and $\widetilde{d_{\mathrm{SV}}}=d_A[\widetilde{A}]+d_B[\widetilde{B}]+d_C[\widetilde{C}]+d_D[\widetilde{D}]$, where $[\xi]=\lambda_\xi[\widetilde{\xi}]$ for $\xi=A,B,C,D$. It becomes apparent that $\tau(a_0+d_0)\ll\tau(\widetilde{a_{\mathrm{SV}}}+\widetilde{d_{\mathrm{SV}}})$.

In principle, we can approximate the original time scales by multiplying each variable in the reaction propensities and diffusion propensities by its corresponding λ_ξ, which consequently leads to $\tau(a_0+d_0)\sim\tau(\widetilde{a_{\mathrm{SV}}}+\widetilde{d_{\mathrm{SV}}})$. We pursue this idea with the understanding that, as a system is divided into SVs, there will be a larger number of diffusion events. However, the addition of these smaller yet more numerous diffusion steps should be similar to their larger yet less numerous counterparts.

In order to explain this, let us assume the biochemical system's domain is initially divided into φ compartments where, in a certain time T^*, a number $\tilde{\varphi}$ of diffusion events occur. The number of events depends on both the time T^* and the effective diffusion rate for calculating the propensities $d=d(D,\varphi)$, where D is the molecular diffusion coefficient. Similarly, if the same biochemical system is divided into ψ compartments, a number $\tilde{\psi}$ of diffusion events will occur, and if $\varphi\ll\psi$, then $\tilde{\varphi}\ll\tilde{\psi}$, since $d(D,\varphi)\ll d(D,\psi)$. However, if we interpret elapsed time as a function of the number of diffusion events, then $\int_0^{\varphi}T(s)ds\sim\int_0^{\psi}T(s)ds$. In other words, if a mean square displacement is covered in average by a single diffusion event, and the latter is split into smaller diffusions steps, these smaller events as a group should take roughly the same time to advance the mean square displacement as the initial single diffusion step.

In order to illustrate these aspects we simulated a single unary reaction with and without time scaling, and computed the relative difference in time the system takes to reach a steady state, for different numbers of SVs. Results are shown in Fig. 3. Additionally, we compared simulation results of unary, heterodimeric, homodimeric, and Hill-type function reactions from NSM, Bτ-SSSA, and CHEMCELL. We found that the temporal dynamics obtained from the NSM and Bτ-SSSA match the stochastic particle simulator result only when time scaling is incorporated.

Finally, it should be mentioned that the parameter ε plays a very special role, especially in scenarios of molecular heterogeneity such as clusters. In these cases, molecules can be so close that potential reactions may happen almost instantaneously, in which case the parameter ε is not only a measure of approximation but also controls the speed in which such reactions within a simulation occur. For example, if we consider a simulation with N clusters containing reacting molecules along with a large value of ε, it will take approximately N time steps to perform all reactions, leaving no room for molecules to diffuse to contiguous SVs. At the

other end of the spectrum, if the value ε is too small, Bτ-SSSA will perform as many steps as the NSM, but in this case the Bτ-SSSA CPU time would be slightly larger than that of NSM, due to computational overhead in calculating the unused values of τ. We suggest that an appropriate value of ε depends on both the number of SVs and the concentration of molecular species. As a rule of thumb, the larger the concentration and/or the number of SVs, the smaller the ϵ that needs to be considered. This topic is current work in progress.

Efficiency: EGFR model

We first test our algorithm on simple sets of reactions of four different types: unary ($A \rightarrow B$), heterodimeric ($A+B \rightarrow C$), homodimeric ($A+A \rightarrow B$), and Hill-function type ($A \rightarrow A+B$). To test the binary reaction types, we simulate a simplified model of progesterone transcription factor formation in the epidermal growth factor receptor (EGFR) pathway.[17,18] In this model, MAPK kinase and progesterone receptor PrR type B are phosphorylated inside the cytoplasm (aERK and aPrRB, respectively). Then, inside the nucleus, both phosphorylated progesterone receptor type A, aPrRA and aPrRB, are able to bind to a ligand (p4), to consequently form homodimers or heterodimers, commonly known as transcription factors (TF's).

For the sole purposes of performance and accuracy testing, we considered all the reaction types and cellular processes mentioned above to be inside an arbitrary cubic volume of 8 μm^3, split into either 1, 8, 27, 64, 216, and 1728 SVs. All molecular species were considered to be in concentrations of either 0.359 or 3.59 μM. This is equivalent to having 1728 or 17280 molecules of each species inside 8 fl, respectively. A rate constant of $10^9 M^{-1}$ s^{-1} was used in all binary reactions and a rate constant of 1 s^{-1} in all unary reactions, with a uniform diffusion coefficient of 10^{-9} cm^2/s. All simulation results were averaged over ten independent runs, showing a standard deviation smaller than 5% of their corresponding mean CPU times. An appropriate ε was chosen for each particular case, such that 99% accuracy with respect to NSM simulation outputs was maintained.

Specifically, we split the model into part (1) simulated in one 8 μm^3 section of the cytoplasm and part (2) simulated in one 8 μm^3 section of the nucleus:

$$\text{aEGFR} + \text{ERK} \rightarrow \text{aERK}$$

$$\text{aERK} + \text{PrRB} \rightarrow \text{aPrRB} + \text{ERK} \quad (1)$$

$$\text{aPrRB} + \text{p4} \rightarrow \text{aPrRBp}$$

$$\text{aPrRBp} + \text{aPrRBp} \rightarrow \text{TF} \quad (2)$$

A homogeneous distribution of molecules throughout the volume was initially assumed for the simulation of reaction set (1) as, to the best of our knowledge, there is no reported evidence suggesting a particular molecular distribution associated to the phosphorylation of PrRB events. The results for reaction set (1) are shown in Table III. Note that all CPU times are referred as multiples of the NSM method's CPU time in one SV with low concentrations of molecules, until a steady state is reached.

TABLE III. Comparison of CPU times in NSM and $B\tau$-SSSA, first reaction set, uniform distribution of molecules as initial condition.

	1728 molecules for each species			17280 molecules for each species	
SVs	CPU time NSM		CPU time $B\tau$-SSSA	CPU time NSM	CPU time $B\tau$-SSSA
1	1	◇	0.18	43.57	0.21
8	1.05		0.21	44.26	0.25
27	1.18		0.26	45.42	0.31
64	1.36		0.39	47.36	0.47
125	1.56		0.68	52.36	0.92
216	2.84		0.98	57.14	1.43

For reaction set (2), we did not consider a uniform distribution of molecules, as PrRB has frequently been observed to be clustered in endometrial cancerous cells.[19,20] However, and to the best of our knowledge, there are no studies indicating whether clusters of aERK, PrRB, and p4 molecules are spatially close or not. Hence, two distinct cluster scenarios were considered.

(a) Cold spots. Molecular species are clustered, but molecular species that can react are separated from each other, i.e., their initial clustering is mutually exclusive.
(b) Hot spots. Molecular species that can react are located within the same cluster.

As before, CPU times are averaged over ten independent runs, showing a standard deviation smaller than 5% of their corresponding mean CPU times. Once again, an appropriate ε was chosen for each particular case, where 99% accuracy with respect to NSM simulation outputs was maintained. For the "cold spots" case each SV contained a cluster of only one reactant molecular species. On the other hand, in the "hot spot case," each one out of four SVs are chosen to contain clusters of all molecular species as an initial condition, while the remaining SVs are empty.

CPU times results for part (2), having "hot spots" as an initial condition, are shown in Table IV. All CPU times are referred to as multiples of the NSM method's CPU time used to calculate set of reactions (2) in one SV with low concentrations of molecules, until the system reaches a steady state.

TABLE IV. Comparison of CPU times in NSM and $B\tau$-SSSA, second reaction set, clusters of molecules ("hotspots") as initial condition.

	1728 molecules for each species			17280 molecules for each species	
SVs	CPU time NSM		CPU time $B\tau$-SSSA	CPU time NSM	CPU time $B\tau$-SSSA
1	1	○	0.12	43.62	0.20
8	1.07		0.15	45.21	0.21
27	1.19		0.18	49.46	0.24
64	1.63		0.20	55.89	0.26
125	2.22		0.38	62.21	0.43
216	6.27		0.69	69.57	0.82

It should be noted that no significant difference in CPU times between both methods was found in the "cold spot" case. This was to be expected, as many single diffusion events will initially dominate the dynamics. It may be worth noting that such simulation times were in average ten times longer than the hot spot case and as is the case for clusters in general, these numbers may be optimized by fine-tuning the parameter ε.

For the particular cases of unary reactions and Hill-type function reactions, a single reaction test was performed, reporting significant differences between the CPU times of Bτ-SSSA and NSM. The simulations of a single unary reaction until reaching steady state, with an initial concentration of 0.359 μM in 8 fl, a reaction rate of 1 s^{-1}, and a diffusion coefficient of 10^{-9} cm^2/s were, on average, 2–40 times faster with Bτ-SSSA, depending on the concentrations.

On the other hand, for a Hill-type function reaction $A \rightarrow A+B$ (described in Table II), simulations were performed until B reached 20% of the total concentration of A, using a Hill coefficient of 2 and a modulation factor of 100 molecules, along with an initial concentration of 0.359 μM in 8 fl, and a diffusion coefficient of 10^{-9} cm^2/s. The results were, on average, 100–200 times faster with Bτ-SSSA. It should be noted that greater CPU time savings are expected to be achieved with larger concentrations.

Accuracy: Bistable model

In order to illustrate accuracy, we compared the NSM and the Bτ-SSSA algorithms on slightly more complicated systems. Out of consistency, we decided to report results on the exact same bistable system used in Ref. 1, with $[E_A]_{\text{tot}}=[E_B]_{\text{tot}}=12.3$ nM, a volume of 27 μm^3, a diffusion rate of 10^{-9} cm^2/s, and reaction rates of $k_l=150$ s^{-1}, $k_a=1.2\times10^8$ s^{-1} M^{-1}, $k_d=10$ s^{-1}, and $k_4=6$ s^{-1} in the limit of fast diffusion.

$$E_A \xrightarrow{k_1} E_A + A \qquad E_B \xrightarrow{k_1} E_B + B$$

$$E_A + B \underset{k_d}{\overset{k_a}{\leftrightarrow}} E_AB \qquad E_AB + B \underset{k_d}{\overset{k_a}{\leftrightarrow}} E_AB_2 \qquad A \xrightarrow{k_4} \varnothing$$

$$E_B + A \underset{k_d}{\overset{k_a}{\leftrightarrow}} E_BA \qquad E_BA + A \underset{k_d}{\overset{k_a}{\leftrightarrow}} E_BA_2 \qquad B \xrightarrow{k_4} \varnothing$$

In the deterministic solution, it can be observed that both species A and B reach the same steady state. However, in the stochastic setting, species A and B can reach different equilibrium states. As a sample, we obtained averages of simulations in each scenario with total time of 3 s and portray the difference of the simulation methods in Fig. 4. It should be noted that the binary rate k_a was modified to account for diffusion dependence.

Additionally, we obtained comparison CPU times averaged over ten independent runs, where approximately 95% accuracy with respect to NSM simulation outputs was maintained. Results are shown in Fig. 5. For this particular case, Bτ-SSSA yielded CPU times five to ten times less than those of NSM.

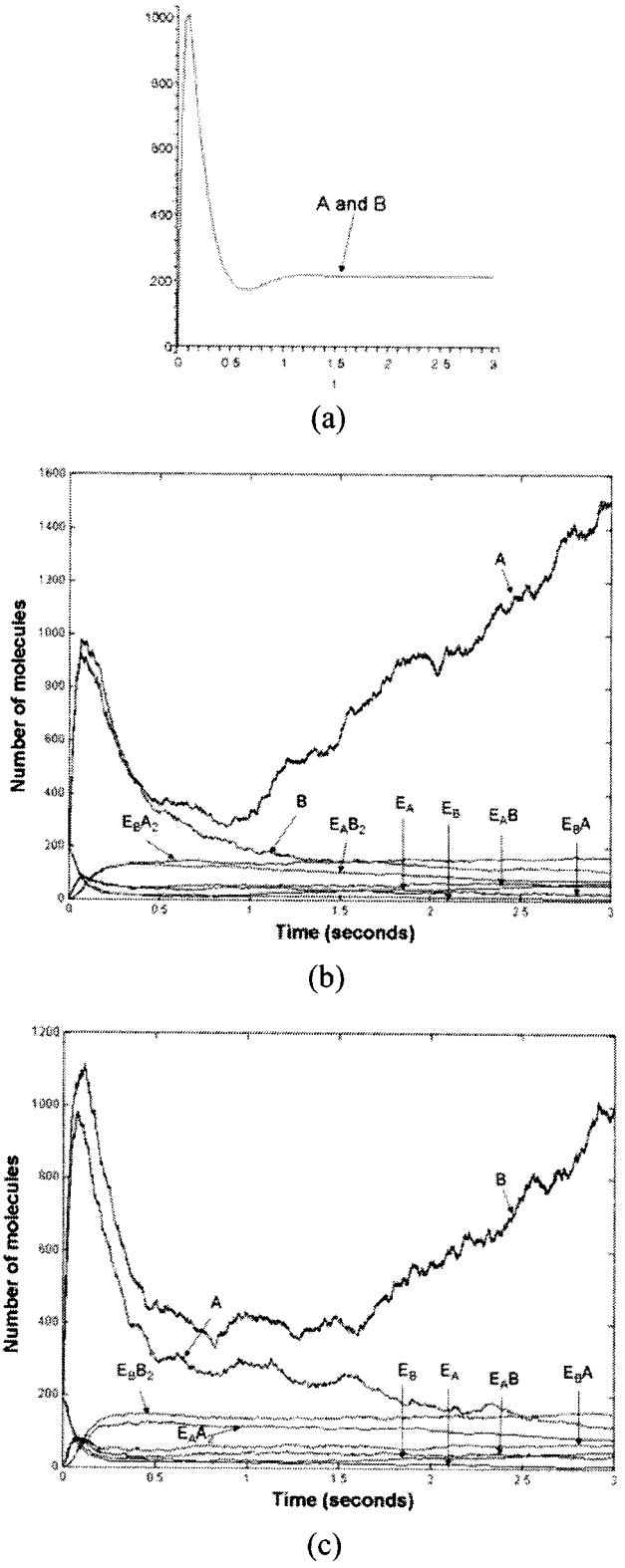

FIG. 4. Solutions of the bistable system. (a) Deterministic solutions for molecular species A and B. (b) Sample single run using the NSM in 64 subvolumes, portraying the case where $A(3)>B(3)$. (c) Sample single run using the NSM in 64 subvolumes, portraying the case where $A(3)<B(3)$.

DISCUSSION

For the reaction sets considered and the described domain partitions, Bτ-SSSA captured all spatial information 3–200 times faster than the NSM (depending on the configu-

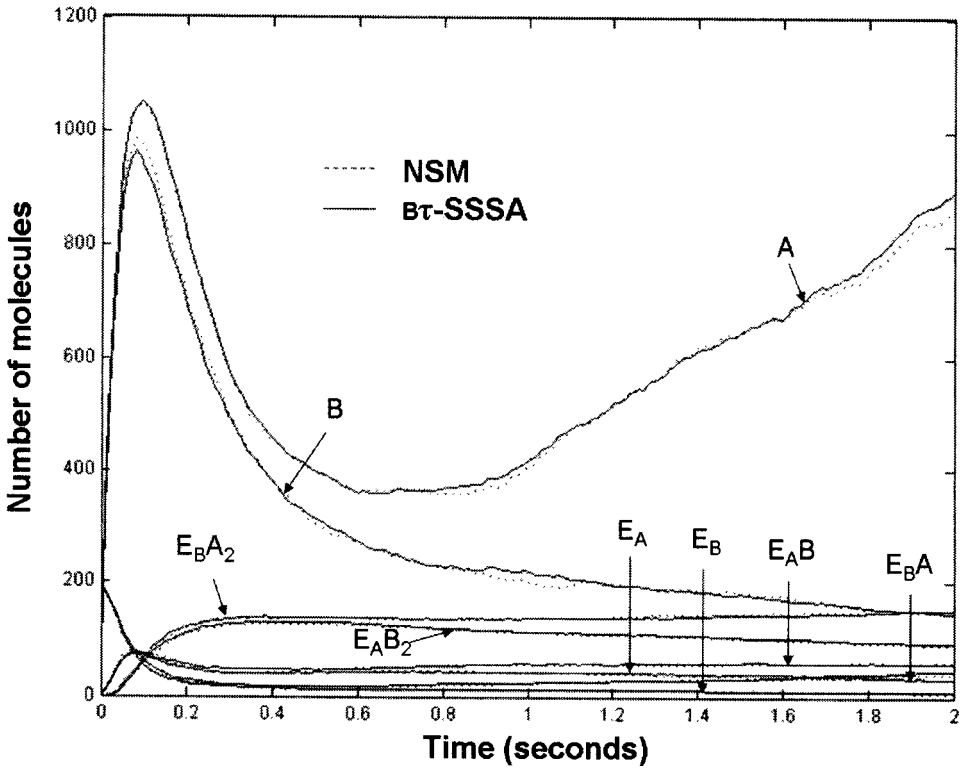

FIG. 5. Average over ten independent runs in which $A > B$ after 2 s, for both NSM and $B\tau$-SSSA in 64 subvolumes (with $\varepsilon = 0.01$).

ration) without compromising accuracy. Most interestingly, it could be seen that, as the number of SVs increases (but is still far away from the KMC end of the spectrum), Bτ-SSSA is most efficient, providing an efficient way to capture spatial details in long simulation time spans. The most striking difference between Bτ-SSSA and the NSM is in the particular case of molecular clustering where, even for low simulation volume subdivision, CPU times range between 85 and 218 faster than the original method. Other domain partitions are expected to yield larger computational savings as compared to the original NSM.

Additionally, on the presented models Bτ-SSSA was approximately 20–120 times faster than the particle off-lattice simulator CHEMCELL at lower molecular concentrations. In scenarios with higher concentrations, Bτ-SSSA was approximately 150–1000 times faster. In CHEMCELL, the resolution is better than Bτ-SSSA, as particle specific coordinates can be obtained at any time step. However, SVs in the Bτ-SSSA can be designed to be small enough to provide with a graphical representation that is close to particle based simulators, while yielding outputs in meaningful real-time spans, and thus promote a better understanding of cell signaling pathways. It should also be noted that CHEMCELL is developed in C++, whereas our comparison algorithms are currently programed in MATLAB and, so, higher computational savings are expected as compared to particle-based simulators when implementing our method in C++.

The bistable biochemical system case is of particular interest, as the concentration inside each SV is very low and demonstrates that Bτ-SSSA is not only accurate with respect to NSM, but is also an efficient tool, since the overhead of drawing random numbers from the binomial distribution is only necessary if the number of molecules inside each SV is large enough. A second filter for a τ step is checking whether the calculated τ is larger than the sum of single even times for ρ events, where ρ is a user-specified lower limit on the number of events within a τ step. Under this light, it becomes obvious that the overhead of calculating steps 10, 11, and 15 within Bτ-SSSA is an "expense" that we will only incur if and only if the system's concentration is large enough. In short words, Bτ-SSSA is expected to be roughly equal or faster than NSM.

Lastly, it should be noted that the performance analysis in this paper applies only to the models presented and, hence, larger computational savings can be expected under higher molecular concentrations, as well as in different cell signaling systems. Very importantly, obtained CPU times are independent of time scaling.

Future extensions of the Bτ-SSSA algorithm, such as permeability between compartments, nonuniform domain partitions, alternative shapes of SVs (e.g., noncubic polyhedra), and anomalous diffusion, are current work in progress.

ACKNOWLEDGMENTS

One of the authors (K.B.) would like to thank the Australian Research Council for its support via the Federation Fellowship program. The authors would also like to give special thanks to Dr. André Leier for his continuous help and input throughout the development of this work.

[1] K. Burrage, M. Hegland, S. MacNamara, and R. B. Sidje, *Proceedings of the Markov 150th Anniversary Conference*, edited by A. N. Langville and W. J. Stewart (Boson Books, Raleigh, NC, 2006), pp. 21–38.
[2] A. Bortz, M. Kalos, and J. Lebowitz, J. Comput. Phys. **17**, 10 (1975).
[3] D. Gillespie, J. Phys. Chem. **81**, 2340 (1977).
[4] M. A. Gibson and J. Bruck, J. Phys. Chem. **104**, 1876 (2000).
[5] D. Gillespie, J. Chem. Phys. **115**, 1716 (2001).
[6] D. Gillespie and L. Petzold, J. Chem. Phys. **119**, 8229 (2003).
[7] T. Tianhai and K. Burrage, J. Chem. Phys. **121**, 10356 (2004).

[8] T. Turner, S. Schnell, and K. Burrage, Comput. Biol. Chem. **28**, 165 (2004).
[9] C. J. Morton-Firth and D. Bray, J. Theor. Biol. **192**, 117 (1998).
[10] D. V. Nicolau, Jr., K. Burrage, R. G. Parton, and J. F. Hancock, Mol. Cell. Biol. **26**, 313 (2006).
[11] S. Plimpton and A. Slepoy, ChemCell: A Particle-Based Model of Protein Chemistry and Diffusion in Microbial Cells, Sandia National Laboratories, Technical Report SAND2003–4509 (2003).
[12] http://www.cs.sandia.gov/~sjplimp/chemcell.html
[13] http://www.mcell.cnl.salk.edu/
[14] J. Elf and M. Ehrenberg, Systems Biology **2**, 230 (2004).
[15] J. Elf, A. Doncic, and M. Ehrenberg, Proc. SPIE **5110**, 114 (2003).
[16] J. Hattne, D. Fange, and J. Elf, Bioinformatics **21**, 2923 (2005).
[17] T. Marquez-Lago, S. Steinberg, A. Slepoy, B. S. Wilson, and K. K. Leslie, "Numerical estimation of ERK-mediated clustered progesterone transcription factor formation" (unpublished).
[18] T. Marquez-Lago, Ph.D. dissertation, University of New Mexico.
[19] M. Qiu, A. Olsen, E. Faivre, K. B. Horwitz, and C. A. Lange, Mol. Endocrinol. **17**, 628 (2003).
[20] R. Arnett-Mansfield, A. DeFazio, P. Mote, and C. Clarke, J. Clin. Endocrinol. Metab. **89**, 1429 (2004).